Brain Computer Interaction

Performance Evaluation

Course Instructors

Dr. Annushree Bablani

Acknowledgements: Dr. Sreeja S R

Introduction

- A classifier is used to predict an outcome of a test data
  - Such a prediction is useful in many applications
    - Business forecasting, cause-and-effect analysis, etc.
  - A number of classifiers have been evolved to support the activities.
    - Each has their own merits and demerits
- There is a need to estimate the accuracy and performance of the classifier with respect to few controlling parameters in data sensitivity

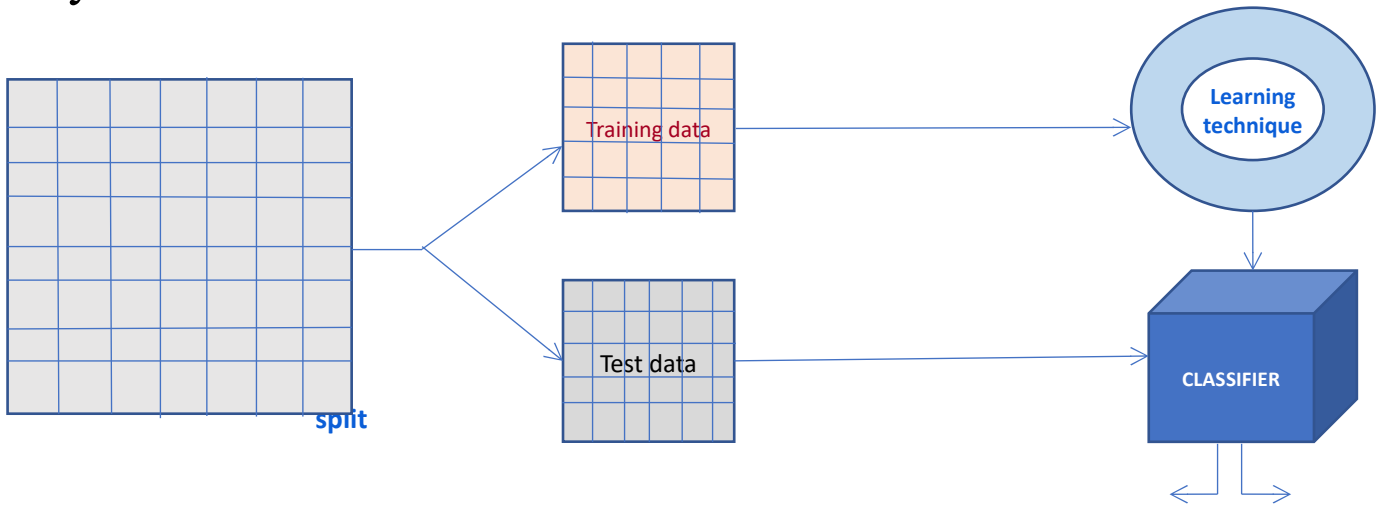
- As a task of sensitivity analysis, we have to focus on

  - Estimation strategy
  - Metrics for measuring accuracy
  - Metrics for measuring performance

Estimation Strategy

Planning for Estimation

- Using some "training data", building a classifier based on certain principle is called "learning a classifier".
- After building a classifier and before using it for classification of unseen instance, we have to validate it using some "test data".
- Usually training data and test data are outsourced from a large pool of data already available.



Estimation Strategies

- Accuracy and performance measurement should follow a strategy. As the topic is important, many strategies have been advocated so far. Most widely used strategies are
  - Holdout method
  - Random subsampling
  - Cross-validation
  - Bootstrap approach

Holdout Method

- This is a basic concept of estimating a prediction.
  - Given a dataset, it is partitioned into two disjoint sets called training set and testing set.
  - Classifier is learned based on the training set and get evaluated with testing set.
  - Proportion of training and testing sets is at the discretion of analyst; typically **1:1 or 2:1**, and there is a trade-off between these sizes of these two sets.
  - If the training set is too large, then model may be good enough, but estimation may be less reliable due to small testing set and vice-versa.

Random Subsampling

- It is a variation of Holdout method to overcome the drawback of over-presenting a class in one set thus under-presenting it in the other set and vice-versa.
- In this method, Holdout method is repeated ***k*** times, and in each time, two **disjoint sets are chosen at random** with a predefined sizes.
- Overall estimation is taken as the average of estimations obtained from each iteration.

Cross-Validation

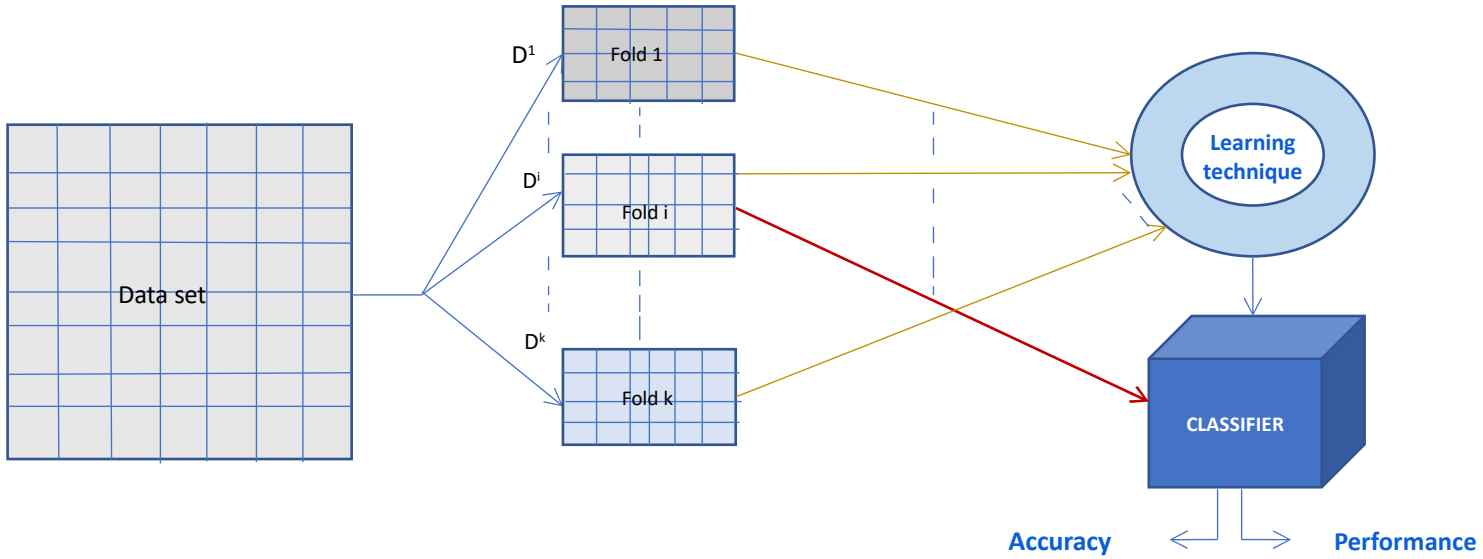
- The main drawback of Random subsampling is, it does not have control over the number of times each tuple is used for training and testing.
- Cross-validation is proposed to overcome this problem.
- There are two variations in the cross-validation method.
  - k-fold cross-validation
  - *N*-fold cross-validation

k-fold Cross-Validation

- Dataset consisting of N tuples is divided into k (usually, 5 or 10) equal, mutually exclusive parts or folds (, and if N is not divisible by k, then the last part will have fewer tuples than other (k-1) parts.

- A series of k runs is carried out with this decomposition, and in i^{th} iteration is used as test data and other folds as training data
  - Thus, each tuple is used same number of times for training and once for testing.

- Overall estimate is taken as the average of estimates obtained from each iteration.



N-fold Cross-Validation

- In *k*-fold cross-validation method, part of the given data is used in training with *k*-tests.

- *N*-fold cross-validation is an extreme case of *k*-fold cross validation, often known as "Leave-one-out" cross-validation.

- Here, dataset is divided into as many folds as there are instances; thus, all most each tuple forming a training set, building *N* classifiers.
- In this method, therefore, *N* classifiers are built from *N-1* instances, and each tuple is used to classify a single test instances.
- Test sets are mutually exclusive and effectively cover the entire set (in sequence). This is as if trained by entire data as well as tested by entire data set.
- Overall estimation is then averaged out of the results of *N* classifiers.

N-fold Cross-Validation : Issue

- So far the estimation of accuracy and performance of a classifier model is concerned, the *N*-fold cross-validation is comparable to the others we have just discussed.

- The drawback of *N*-fold cross validation strategy is that it is computationally expensive, as here we have to repeat the run *N* times; this is particularly true when data set is large.

- In practice, the method is extremely beneficial with very small data set only, where as much data as possible to need to be used to train a classifier.

Bootstrap Method

- The Bootstrap method is a variation of repeated version of Random sampling method.
- The method suggests the sampling of training records with replacement.
  - Each time a record is selected for training set, is put back into the original pool of records, so that it is equally likely to be redrawn in the next run.
  - In other words, the Bootstrap method samples the given data set uniformly with replacement.
- The rational of having this strategy is that let some records be occur more than once in the samples of both training as well as testing.
  - What is the probability that a record will be selected more than once?

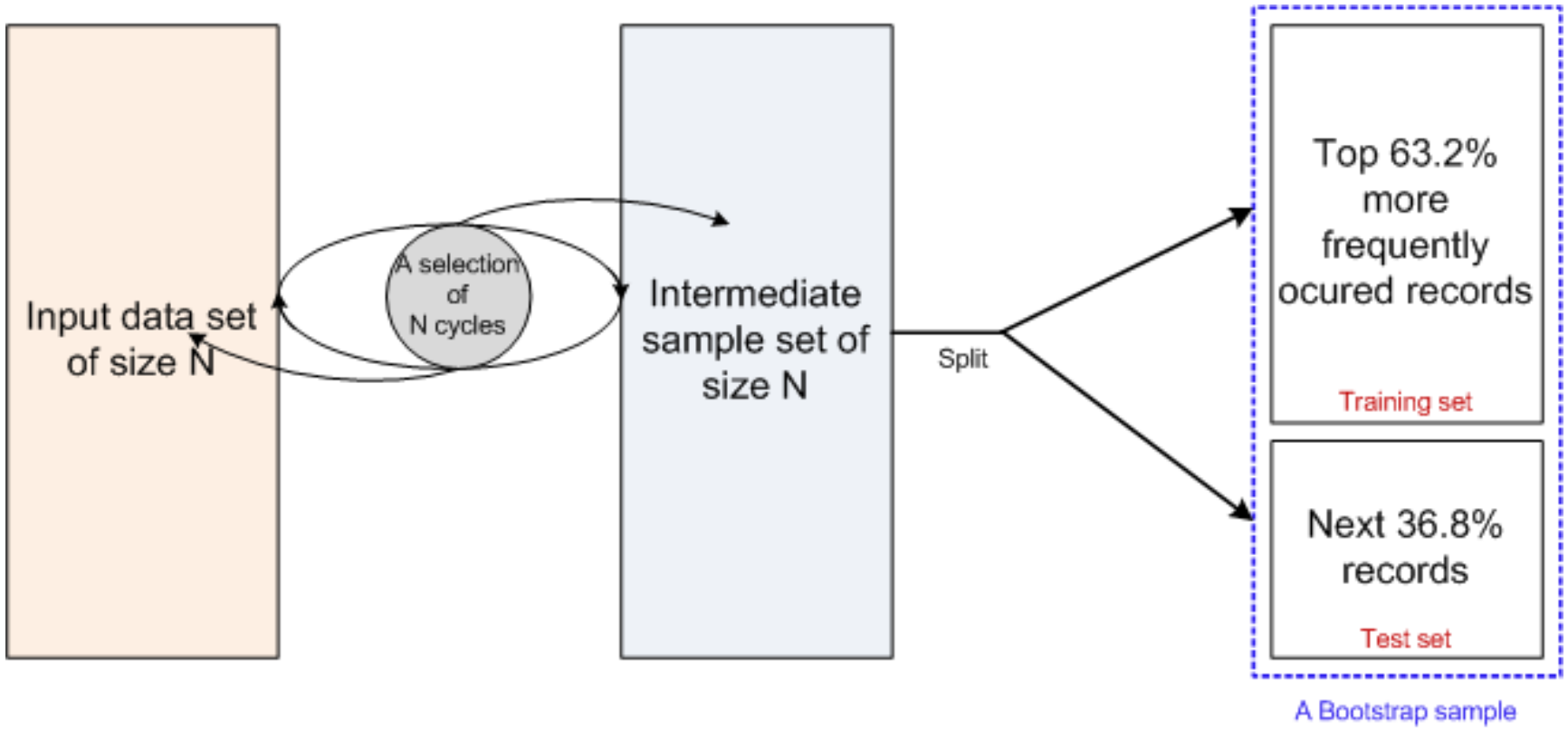
Bootstrap Method

- Suppose, we have given a data set of *N* records. The data set is sampled *N* times with replacement, resulting in a bootstrap sample (i.e., training set) of I samples.
  - Note that the entire runs are called a bootstrap sample in this method.

- There are certain chance (i.e., probability) that a particular tuple occurs one or more times in the training set

  - If they do not appear in the training set, then they will end up in the test set.
  - Each tuple has a probability of being selected (and the probability of not being selected is .
  - We have to select N times, so the probability that a record will not be chosen during the whole run is
  - Thus, the probability that a record is chosen by a bootstrap sample is
  - For a large value of *N*, it can be proved that
  - record chosen in a bootstrap sample is = 0.632

Bootstrap Method : Implication



- This is why, the Bootstrap method is also known as 0.632 bootstrap method

Accuracy Estimation

Accuracy Estimation

- We have learned how a classifier system can be tested. Next, we are to learn the metrics with which a classifier should be estimated.

- There are mainly two things to be measured for a given classifier
  - Accuracy
  - Performance

- **Accuracy estimation**
  - If N is the number of instances with which a classifier is tested and p is the number of correctly classified instances, the accuracy can be denoted as

  - Also, we can say the error rate (i.e., is misclassification rate) denoted by is denoted by

-

Accuracy : True and Predictive

- Now, this accuracy may be true (or absolute) accuracy or predicted (or optimistic) accuracy.
- True accuracy of a classifier is the accuracy when the classifier is tested with all possible unseen instances in the given classification space.
  - However, the number of possible unseen instances is potentially very large (if it is not infinite)
  - For example, classifying a hand-written character
  - Hence, measuring the true accuracy beyond the dispute is impractical.
- Predictive accuracy of a classifier is an accuracy estimation for a given test data (which are mutually exclusive with training data).
  - If the predictive accuracy for test set is and if we test the classifier with a different test set it is very likely that a different accuracy would be obtained.
  - The predictive accuracy when estimated with a given test set it should be acceptable without any objection

Performance Estimation

Performance Estimation of a Classifier

- Predictive accuracy works fine, when the classes are balanced
  - That is, every class in the data set are equally important
- In fact, data sets with imbalanced class distributions are quite common in many real life applications
- When the classifier classified a test data set with imbalanced class distributions then, predictive accuracy on its own is not a reliable indicator of a classifier's effectiveness.

Example 22.1: Effectiveness of Predictive Accuracy

- Given a data set of stock markets, we are to classify them as "good" and "worst". Suppose, in the data set, out of 100 entries, 98 belong to "good" class and only 2 are in "worst" class.
  - With this data set, if classifier's predictive accuracy is 0.98, a very high value!
    - Here, there is a high chance that 2 "worst" stock markets may incorrectly be classified as "good"
  - On the other hand, if the predictive accuracy is 0.02, then none of the stock markets may be classified as "good"

Performance Estimation of a Classifier

- Thus, when the classifier classified a test data set with imbalanced class distributions, then predictive accuracy on its own is not a reliable indicator of a classifier's effectiveness.

- This necessitates an alternative metrics to judge the classifier.

- Before exploring them, we introduce the concept of Confusion matrix.

Confusion Matrix

- A confusion matrix for a two classes (+, -) is shown below.

	C1	C2
C1	True positive	False negative
C2	False positive	True negative

	+	-
+	++	+-
-	-+	--

- There are four quadrants in the confusion matrix, which are symbolized as below.
  - True Positive (TP: f_{++}) : The number of instances that were positive (+) and correctly classified as positive (+v).
  - False Negative (FN: f_{+-}): The number of instances that were positive (+) and incorrectly classified as negative (-).
  - False Positive (FP: f_{-+}): The number of instances that were negative (-) and incorrectly classified as (+).
  - True Negative (TN: f_{--}): The number of instances that were negative (-) and correctly classified as (-).

Confusion Matrix

Example 22.2: Confusion matrix

A classifier is built on a dataset regarding Good and Worst classes of stock markets. The model is then tested with a test set of 10000 unseen instances. The result is shown in the form of a confusion matrix. The result is self explanatory.

Class	Good	Worst	Total
Good	6954	46	7000
Worst	412	2588	3000
Total	**7366**	**2634**	**10000**

Predictive accuracy?

Performance Evaluation Metrics

- We now define a number of metrics for the measurement of a classifier.
  - In our discussion, we shall make the assumptions that there are only two classes: + (positive) and – (negative)
  - Nevertheless, the metrics can easily be extended to multi-class classifiers (with some modifications)

- **True Positive Rate (TPR):** It is defined as the fraction of the positive examples predicted correctly by the classifier.

  =

  - This metrics is also known as ***Recall***, ***Sensitivity*** or ***Hit rate***.

- **False Positive Rate (FPR):** It is defined as the fraction of negative examples classified as positive class by the classifier.

  - This metric is also known as ***False Alarm Rate***.

Performance Evaluation Metrics

- **False Negative Rate (FNR):** It is defined as the fraction of positive examples classified as a negative class by the classifier.

- **True Negative Rate (TNR):** It is defined as the fraction of negative examples classified correctly by the classifier

  - This metric is also known as ***Specificity***.

Performance Evaluation Metrics

- **Positive Predictive Value (PPV):** It is defined as the fraction of the positive examples classified as positive that are really positive
  - It is also known as *Precision*.
- **F_1 Score (F_1):** Recall (r) and Precision (p) are two widely used metrics employed in analysis, where detection of one of the classes is considered more significant than the others.
  - It is defined in terms of (r or TPR) and (p or PPV) as follows.

Predictive Accuracy (ε)

- It is defined as the fraction of the number of examples that are correctly classified by the classifier to the total number of instances.

- This accuracy is equivalent to F_w with $w_1 = w_2 = w_3 = w_4 = 1$.

Error Rate ()

- The error rate is defined as the fraction of the examples that are incorrectly classified.

Note

.

Accuracy, Sensitivity and Specificity

- Predictive accuracy () can be expressed in terms of sensitivity and specificity.
- We can write

Thus,

Analysis with Performance Measurement Metrics

- Based on the various performance metrics, we can characterize a classifier.
- We do it in terms of TPR, FPR, Precision and Recall and Accuracy
- **Case 1: Perfect Classifier**

When every instance is correctly classified, it is called the perfect classifier. In this case, $TP = P$, $TN = N$ and CM is

$TPR = \ = 1$

$FPR = = 0$

$Precision = \ = 1$

$F_1\ Score = = 1$

$Accuracy = \ \ = 1$

		Predicted Class	
		+	-
Actual class	+	P	0
	-	0	N

Analysis with Performance Measurement Metrics

- **Case 2: Worst Classifier**

When every instance is wrongly classified, it is called the worst classifier. In this case, $TP = 0$, $TN = 0$ and the CM is

$TPR = \quad = 0$

$FPR = \quad = 1$

$Precision = \quad = 0$

$F_1\ Score$ = Not applicable as $Recall + Precision = 0$

Accuracy = $\quad = 0$

		Predicted Class	
		+	-
Actual class	+	0	P
	-	N	0

Analysis with Performance Measurement Metrics

- **Case 3: Ultra-Liberal Classifier**

The classifier always predicts the + class correctly. Here, the False Negative (FN) and True Negative (TN) are zero. The CM is

		Predicted Class	
		+	-
Actual class	+	P	0
	-	N	0

$TPR = \ = 1$

$FPR = \ = 1$

$Precision =$

$F_1\ Score =$

Accuracy =

Analysis with Performance Measurement Metrics

- **Case 4: Ultra-Conservative Classifier**

This classifier always predicts the - class correctly. Here, the False Negative (FN) and True Negative (TN) are zero. The CM is

		Predicted Class	
		+	-
Actual class	+	0	p
	-	0	N

$TPR = \quad = 0$

$FPR = \quad = 0$

$Precision =$

(as $TP + FP = 0$)

$F_1\ Score =$

Accuracy =

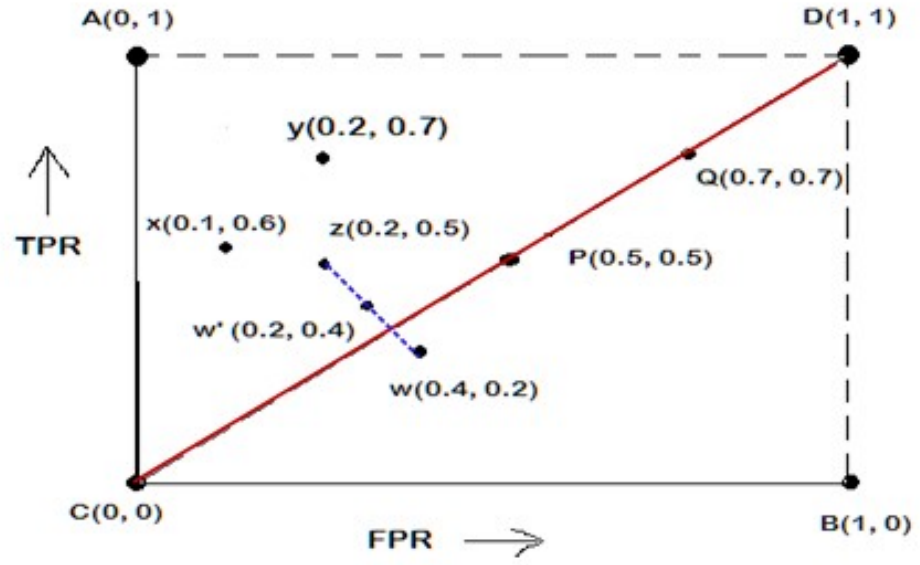
ROC Curves

ROC Curves

- ROC is an abbreviation of Receiver Operating Characteristic come from the signal detection theory, developed during World War 2 for analysis of radar images.
- In the context of classifier, ROC plot is a useful tool to study the behaviour of a classifier or comparing two or more classifiers.
- A ROC plot is a two-dimensional graph, where, X-axis represents FP rate (FPR) and Y-axis represents TP rate (TPR).
- Since, the values of FPR and TPR varies from 0 to 1 both inclusive, the two axes thus from 0 to 1 only.
- Each point *(x, y)* on the plot indicating that the FPR has value *x* and the TPR value *y*.

Interpretation of Different Points in ROC Plot

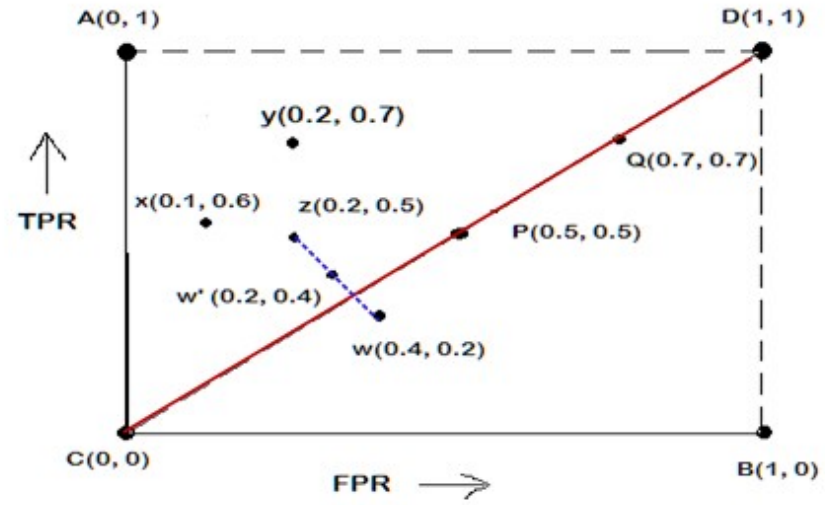
- Let us interpret the different points in the ROC plot.



- The four points (A, B, C, and D)
  - A: TPR = 1, FPR = 0, the ideal model, i.e., the perfect classifier, no false results
  - B: TPR = 0, FPR = 1, the worst classifier, not able to predict a single instance
  - C: TPR = 0, FPR = 0, the model predicts every instance to be a Negative class, i.e., it is an ultra-conservative classifier
  - D: TPR = 1, FPR = 1, the model predicts every instance to be a Positive class, i.e., it is an ultra-liberal classifier

Interpretation of Different Points in ROC Plot

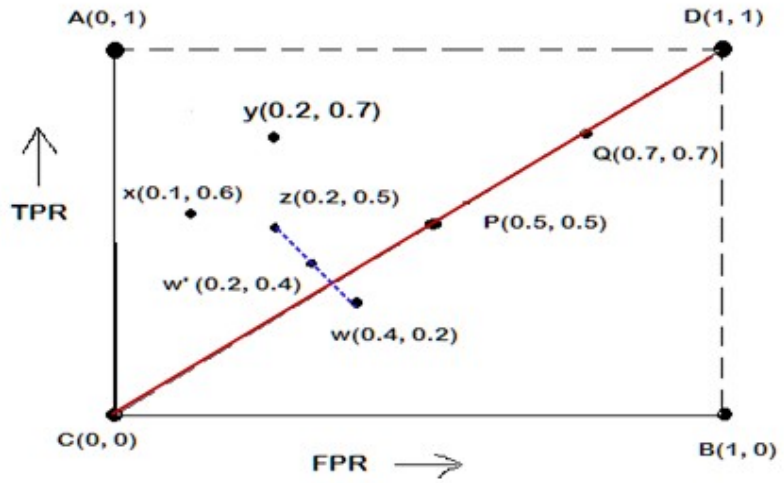
- Let us interpret the different points in the ROC plot.



- The points on diagonals
  - The diagonal line joining point C(0,0) and D(1,1) corresponds to random guessing
    - Random guessing means that a record is classified as positive (or negative) with a certain probability
    - Suppose, a test set contacting N_+ positive and N_- negative instances. Suppose, the classifier guesses any instances with probability p
    - Thus, the random classifier is expected to correctly classify $p.N_+$ of the positive instances and $p.N_-$ of the negative instances
    - Hence, $TPR = FPR = p$
    - Since $TPR = FPR$, the random classifier results reside on the main diagonals

Interpretation of Different Points in ROC Plot

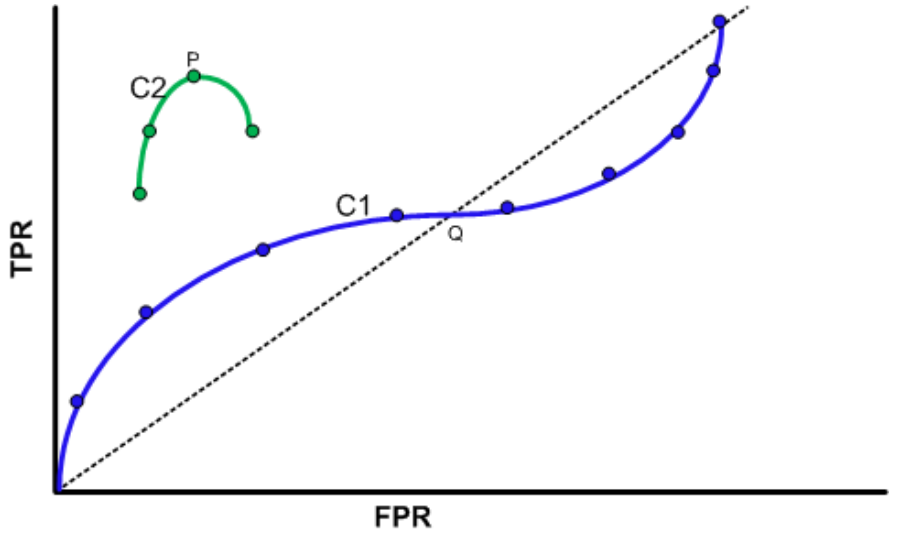
- Let us interpret the different points in the ROC plot.



- The points on the upper diagonal region
  - All points, which reside on upper-diagonal region are corresponding to classifiers "good" as their TPR is as good as FPR (i.e., FPRs are lower than TPRs)
  - Here, X is better than Z as X has higher TPR and lower FPR than Z.
  - If we compare X and Y, neither classifier is superior to the other

Tuning a Classifier through ROC Plot

- Using ROC plot, we can compare two or more classifiers by their TPR and FPR values and this plot also depicts the trade-off between TPR and FPR of a classifier.

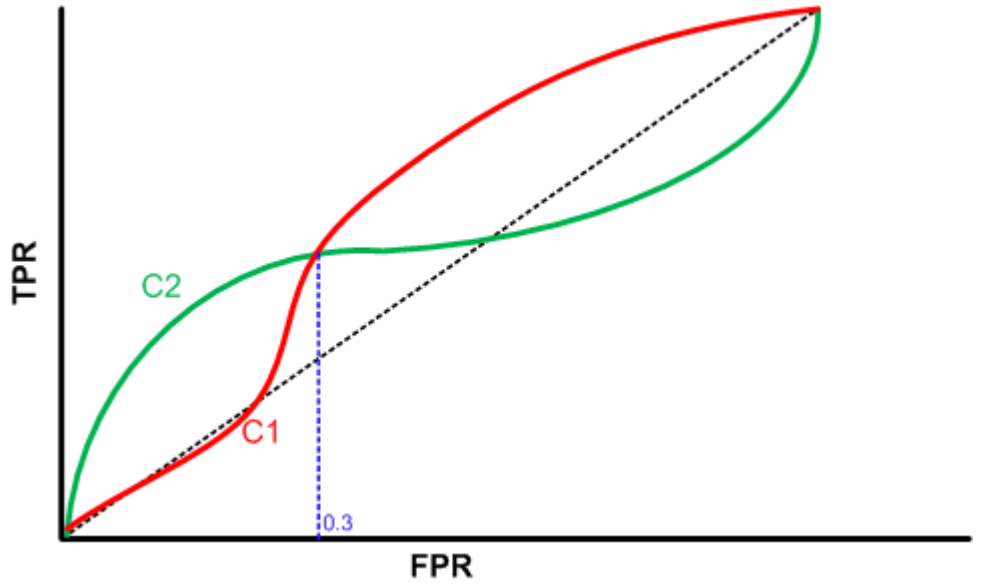


- Examining ROC curves can give insights into the best way of tuning parameters of classifier.

- For example, in the curve C2, the result is degraded after the point P. Similarly for the observation C1, beyond Q the settings are not acceptable.

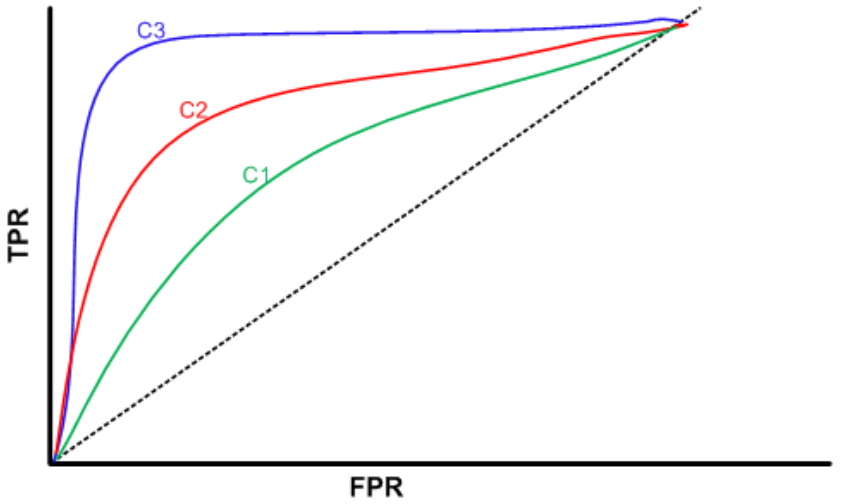
Comparing Classifiers through ROC Plot

- Two curves C1 and C2 are corresponding to the experiments to choose two classifiers with their parameters.
- Here, C1 is better than C2 when FPR is less than 0.3.
- However, C2 is better, when FPR is greater than 0.3.
- Clearly, neither of these two classifiers dominates the other.



Comparing Classifiers trough ROC Plot

- We can use the concept of "area under curve" (AUC) as a better method to compare two or more classifiers.
- If a model is perfect, then its AUC = 1.
- If a model simply performs random guessing, then its AUC = 0.5
- A model that is strictly better than other, would have a larger value of AUC than the other.



- Here, C3 is best, and C2 is better than C1 as AUC(C3)>AUC(C2)>AUC(C1).

Any question?